Lecture 3: Turbo Codes

References

- 16.1-16.5 of Lin's book: Shu Lin and Daniel J. Costello, "**Error Control Coding**," 2nd edition, Prentice Hall, 2004.
- Ch4 and Ch6 of Vucetic's book: B. Vucetic and J. Yuan, "**Turbo Codes: Principles and Applications**," Kluwer Academic Publishers, 2000.
- J. P. Woodward and L. Hanzo, "Comparative study of turbo decoding algorithms: An overview," *IEEE Tran. Vehicular Technology*, vol. 2, no. 5, pp.2208-2233, Nov. 2000.

NSC and RSC

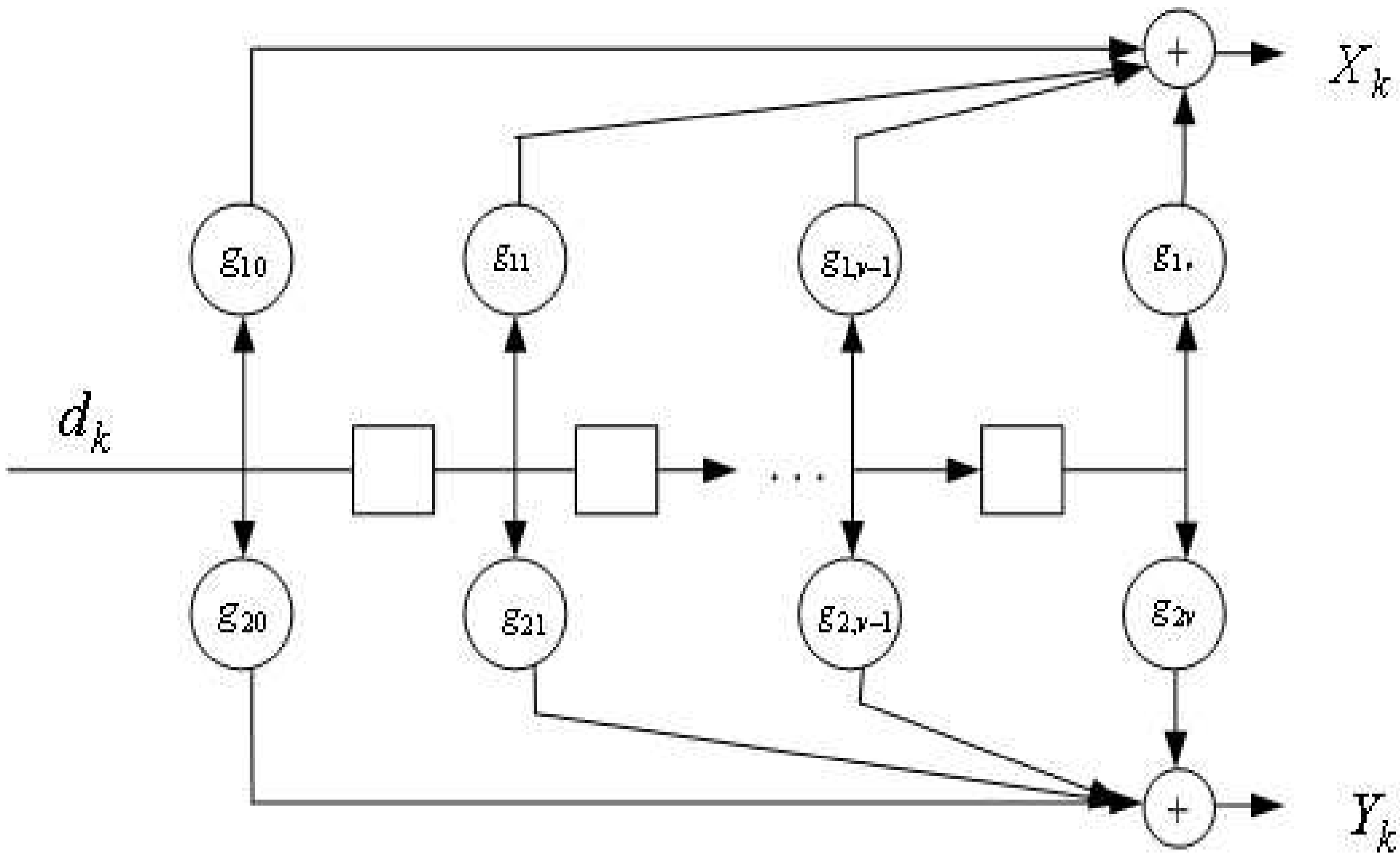
⋆ A nonsystematic convolutional code (NSC) can be converted into recursive systematic convolutional code (RSC) without changing its distance property.

⋆Consider a rate $1/2$ nonsystematic convolutional code with memory size ν and generator sequence $\overline{g}_1 = (g_{10}, g_{11}, g_{12}, \cdots, g_{1\nu})$ and $\overline{g}_2 = (g_{20}, g_{21}, g_{22}, \cdots, g_{2\nu})$ respectively.

⋆Let $\vec{d}$ represent the input sequence and $\vec{X}, \vec{Y}$ represent the two output sequences. We have

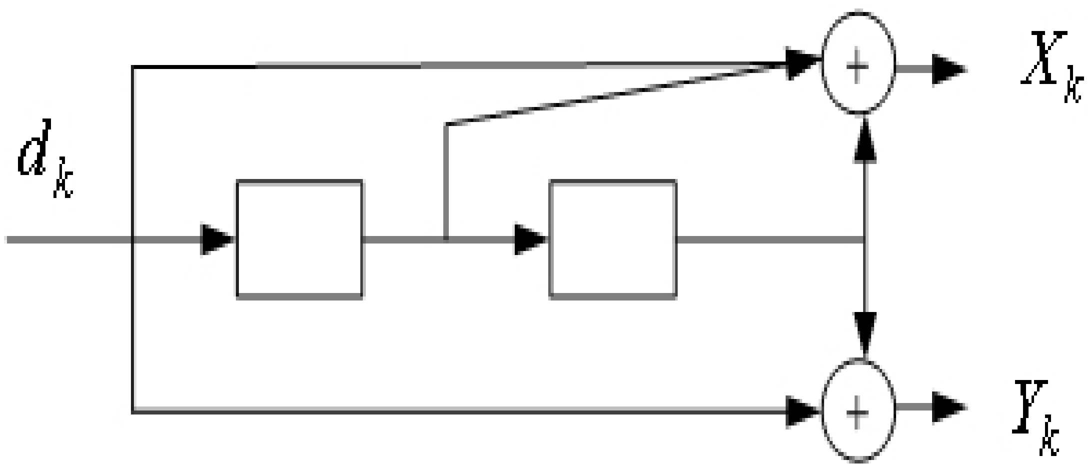
$$X_k = \sum_{i=0}^{\nu} g_{1i} d_{k-i}$$

$$Y_k = \sum_{i=0}^{\nu} g_{2i} d_{k-i}$$



Example 1: Let $\nu = 2, \overline{g_1} = (111), \overline{g_2} = (101)$.

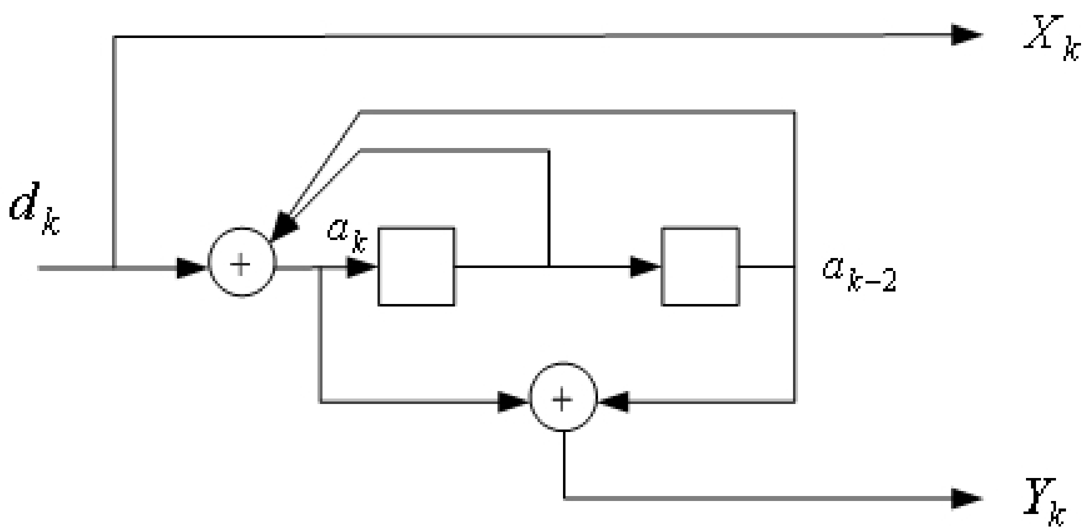
⋆The NSC encoder is



$$X_k = g_{10}d_k + g_{11}d_{k-1} + g_{12}d_{k-2}$$
$$Y_k = g_{20}d_k + g_{21}d_{k-1} + g_{22}d_{k-2}$$

$$G(D) = [1 + D + D^2, 1 + D^2]$$

⋆The RSC encoder is



$$
\begin{aligned}
X_k &= d_k \\
Y_k &= g_{20}a_k + g_{21}a_{k-1} + g_{22}a_{k-2} \\
a_k &= d_k + g_{11}a_{k-1} + g_{12}a_{k-2}
\end{aligned}
$$

$$G(D) = [1, \frac{1 + D^2}{1 + D + D^2}]$$

If $g_{10} = 1$, then

$$X_k = d_k = g_{10} a_k + g_{11} a_{k-1} + g_{12} a_{k-2}$$

In general, let

$$\begin{aligned} X_k &= d_k \\ Y_k &= \sum_{i=0}^{\nu} g_{2i} a_{k-i} \end{aligned}$$

Where a_k is the input to the shift register.

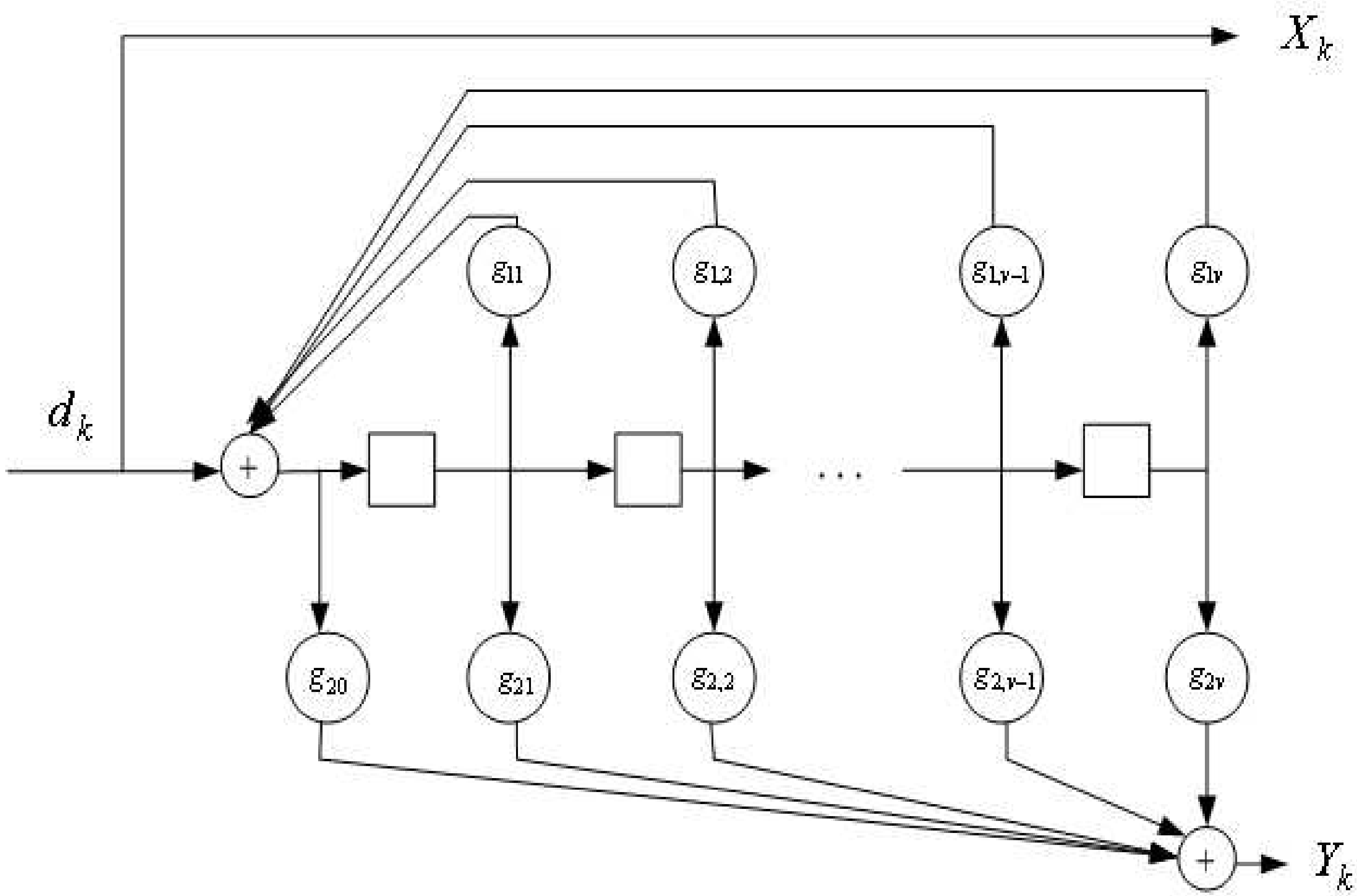
Then,

$$a_k = d_k + \sum_{i=1}^{\nu} g_{1i} a_{k-i}$$

If $g_{10} = 1$, then

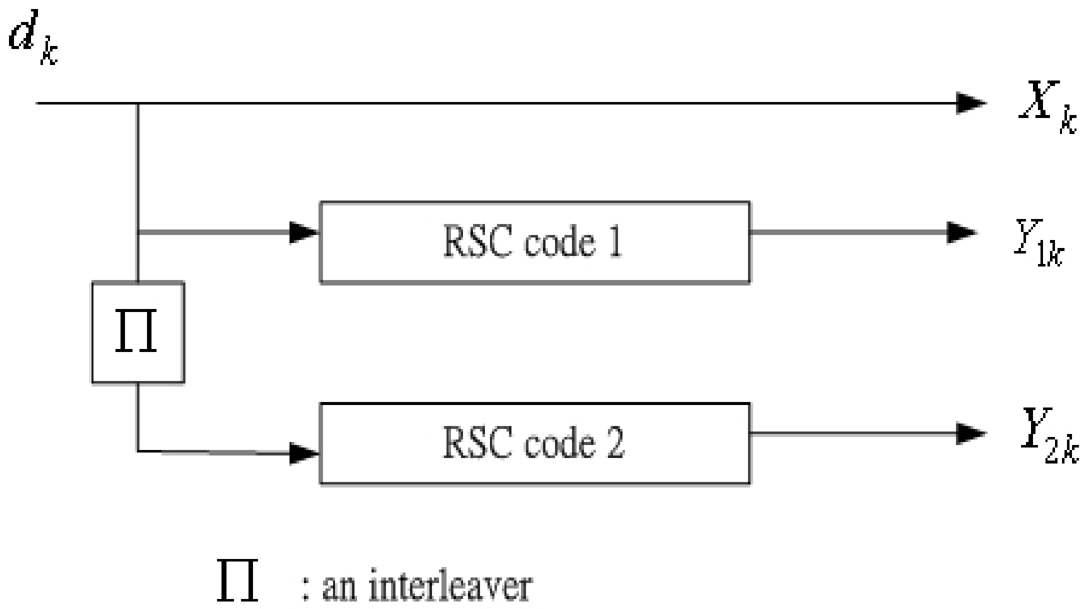
$$X_k = d_k = \sum_{i=0}^{\nu} g_{1i} a_{k-i}$$

⋆The RSC encoder is



⋆The distance properties of NSC and RSC are identical. However, the relationship between input and output is different.

Turbo Encoder

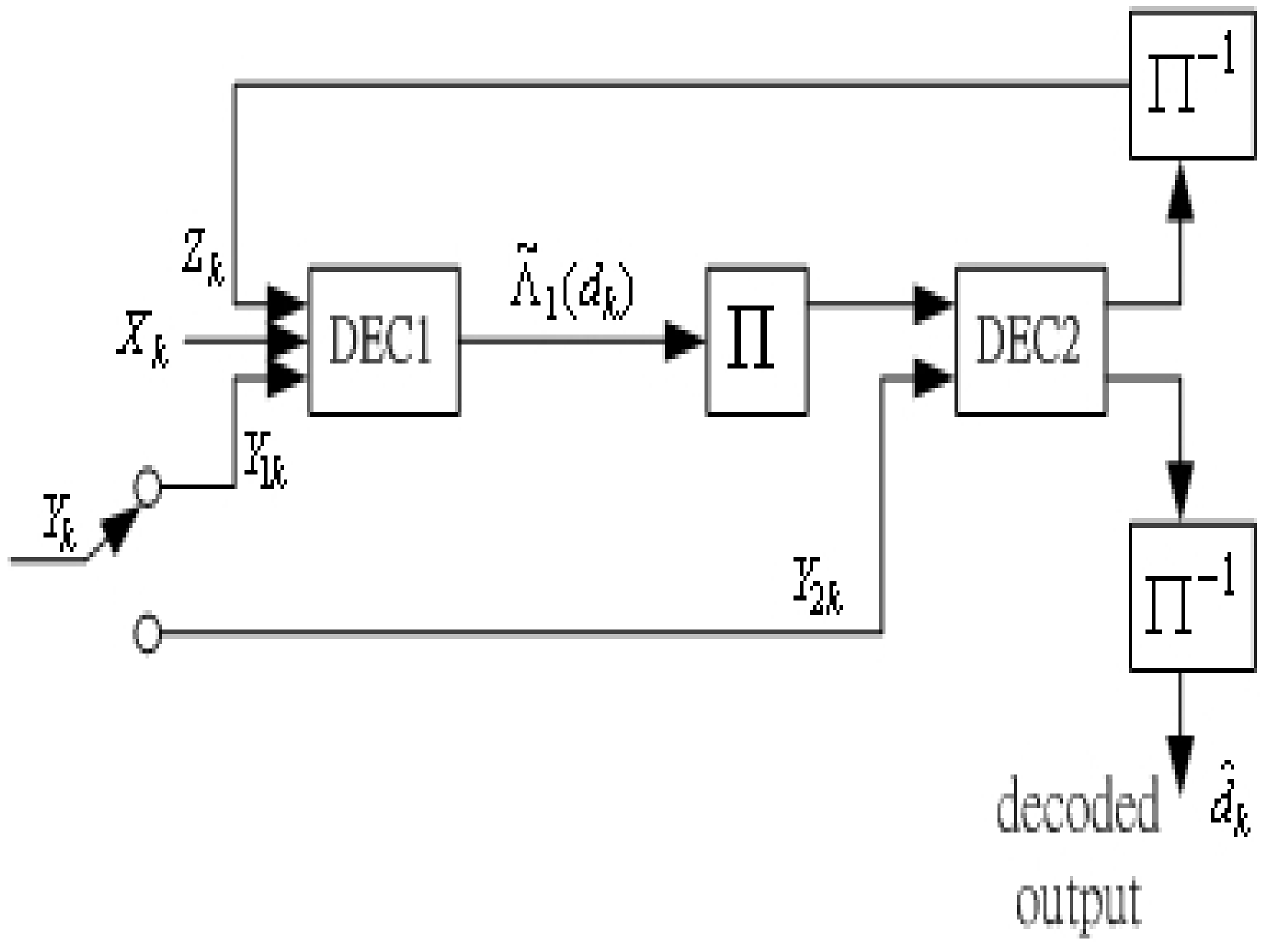


⋆ Let R_1 and R_2 be code rates of RSC code 1 and RSC code 2 respectively. Suppose $R_1 = R_2 = \frac{1}{2}$. The overall code rate will be $R = \frac{1}{3}$.

⋆ By punctuating RSC code 1 and RSC code 2, we can have $R_1 = R_2 = \frac{2}{3}$ and the overall code rate is $R = \frac{1}{2}$.

⋆ Nonuniform interleaving is preferred. Size of interleaver M is critical.

Iterative Decoder



Soft-in/Soft-out Decoder

$L(u)$: a priori values for all information bits.

$L_c y_1$: channel values for all information bits.

$L_e(\hat{u})$: extrinsic values for all information bits.

$L(\hat{u})$: a posteriori values for all information bits.

⋆ For a systematic code, the soft output for an information bit u will be

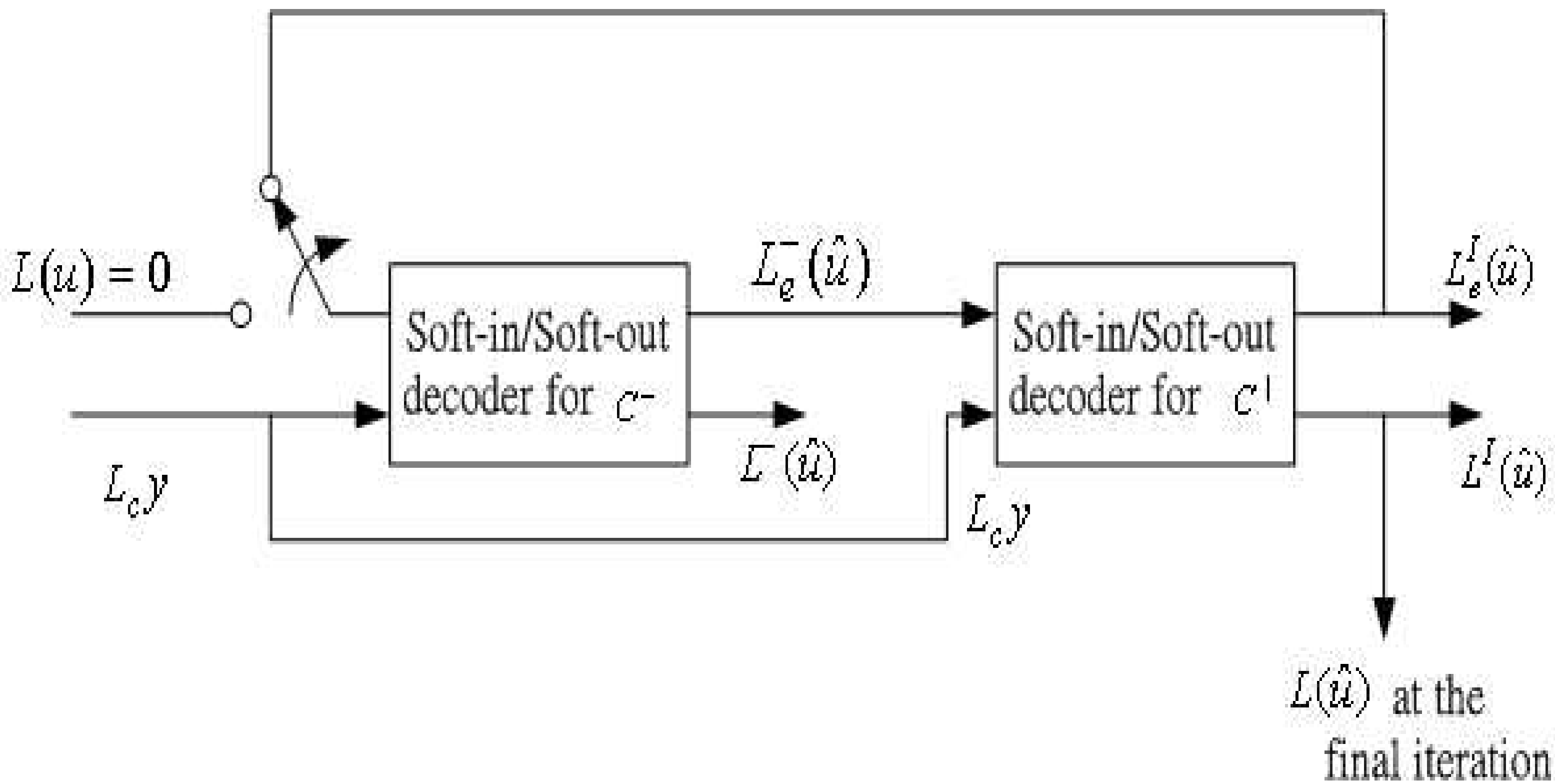
$$\begin{aligned} L(\hat{u}) &= L(u|\bar{y}) = \log \frac{P(u=+1|\bar{y})}{P(u=-1|\bar{y})} \\ &= L_c y_1 + L(u) + L_e(\hat{u}) \end{aligned}$$

Let $\bar{x} = (x_{11}, x_{12}, \cdots, x_{1n}, \cdots, x_{L1}, \cdots, x_{Ln})$ be a codeword of L branches, where $x_{i1} = u_i$ and $x_{i2}, \cdots, x_{in}$ are parity bits.

Let $\bar{y} = (y_{11}, y_{12}, \cdots, y_{1n}, \cdots, y_{L1}, y_{L2}, \cdots, y_{Ln})$ be the received vector. Then, we have

$$
\begin{aligned}
L(\hat{u}_i) &= L(u_i|\bar{y}) = \log \frac{p(u_i = +1|\bar{y})}{P(u_i = -1|\bar{y})} \\
&= \log \frac{P(y_{11}, \cdots, y_{Ln}|u_i = +1)}{P(y_{11}, \cdots, y_{Ln}|u_i = -1)} + \log \frac{P(u_i = +1)}{P(u_i = -1)} \\
&= L_c y_{i1} + \log \frac{P(\bar{y} - \{y_{i1}\}|u_i = +1)}{P(\bar{y} - \{y_{i1}\}|u_i = -1)} + L(u_i) \\
&= L_c y_{i1} + L_e(\hat{u}_i) + L(u_i)
\end{aligned}
$$

Iterative Decoding



Procedure :

(1) $i = 1$. Set $L(u) = 0$.

(2) For the ith iteration, use $L(u)$ and $L_c y$ to calculate $L^{-}(\hat{u})$. Then, calculate the extrinsic information.

$$L_e^{-}(\hat{u}) = L^{-}(\hat{u}) - [L_c y_1 + L(u)]$$

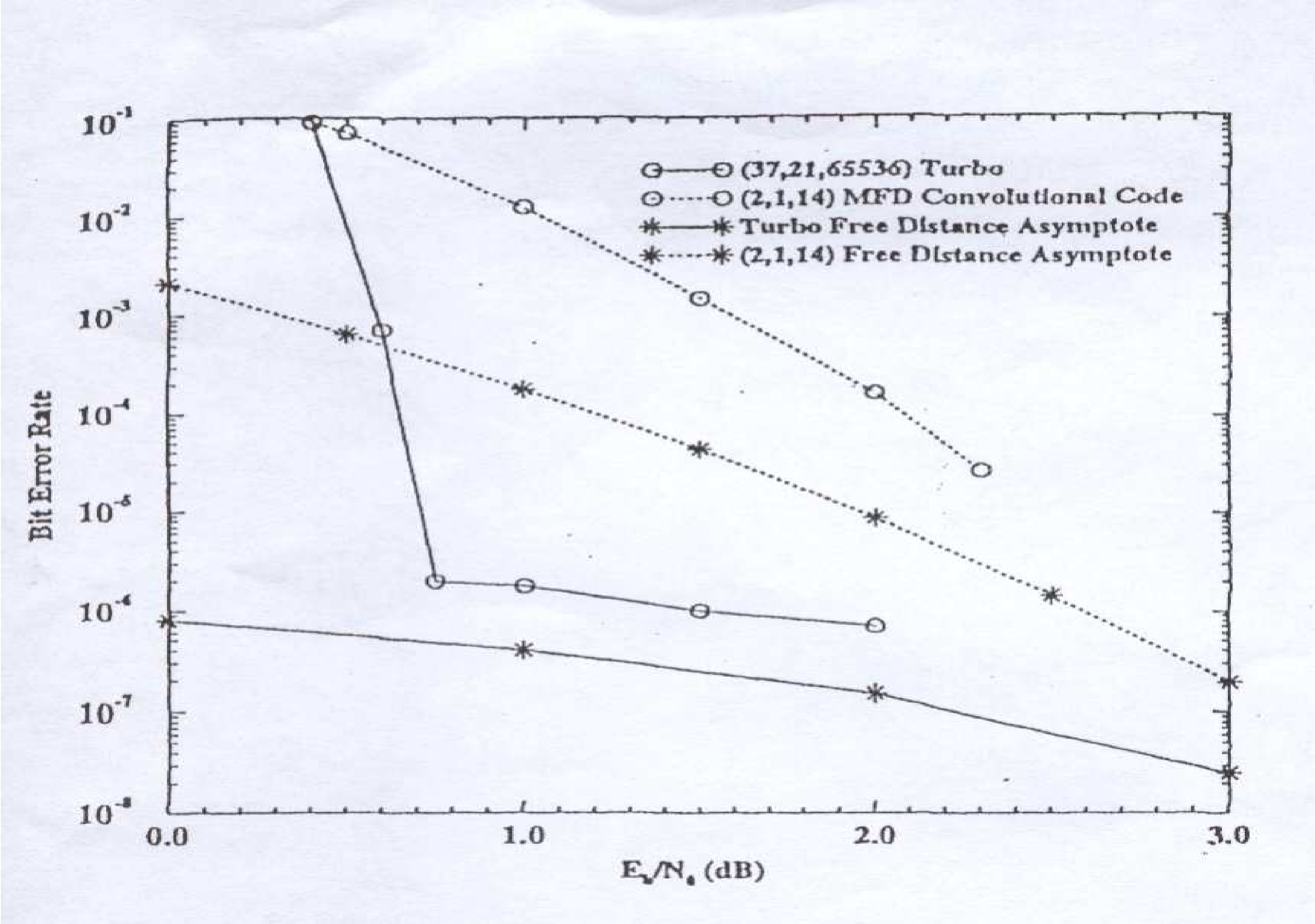
(3) Use $L_e^{-}(\hat{u})$ and $L_c y$ to calculate $L^{|}(\hat{u})$. Then, calculate

$$L_e^{|}(\hat{u}) = L^{|}(\hat{u}) - [L_c y_1 + L_e^{-}(\hat{u})]$$

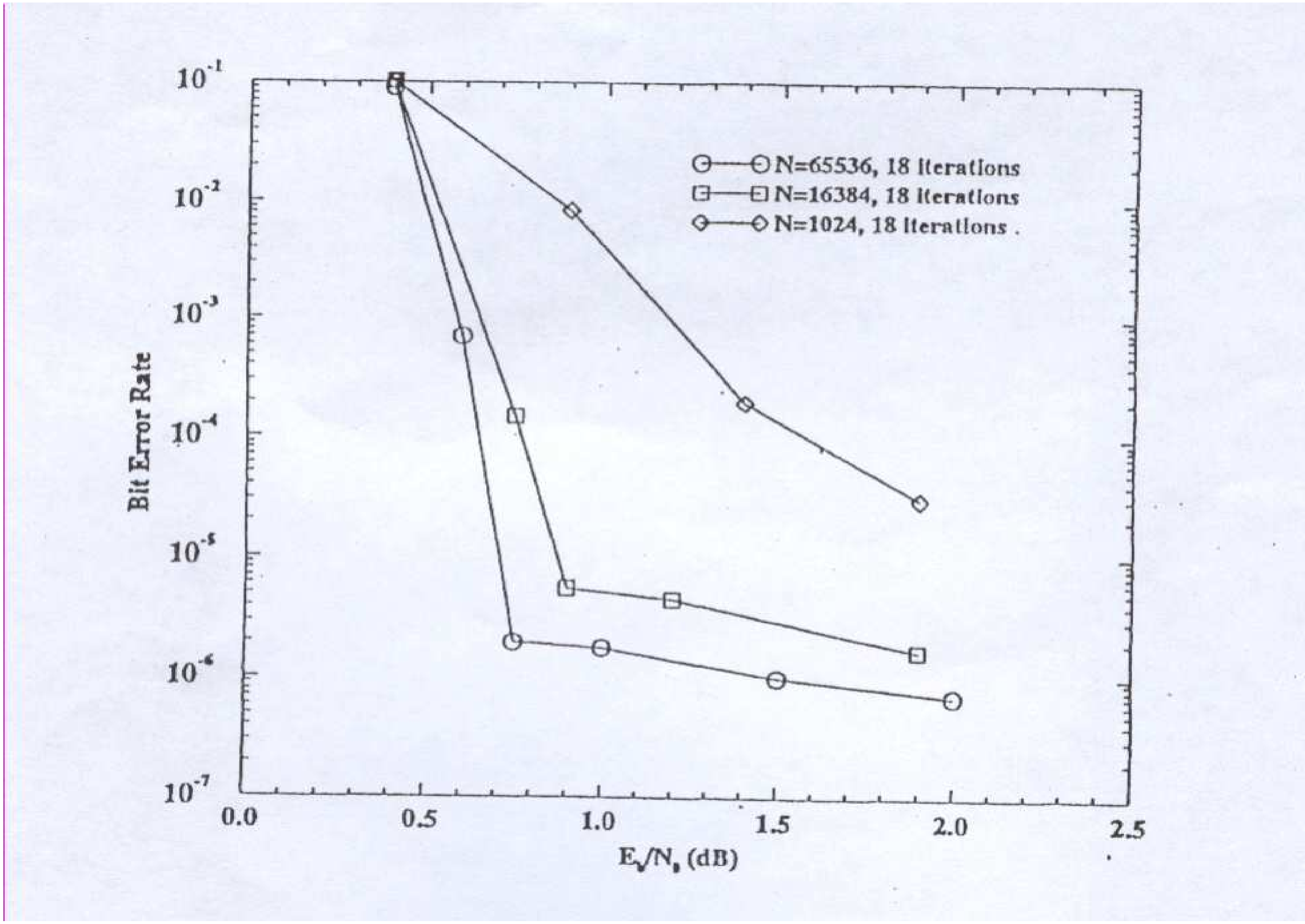
(4) If $i < I$, then $i \rightarrow (i+1)$. Let $L(u) = L_e^{|}(u)$ and go to (2). If $i = I$, then stop. Note that

$$L(\hat{u}) = L_c y + L_e^{-}(\hat{u}) + L_e^{|}(\hat{u})$$

BER Performance: Turbo Codes vs Conv. Codes



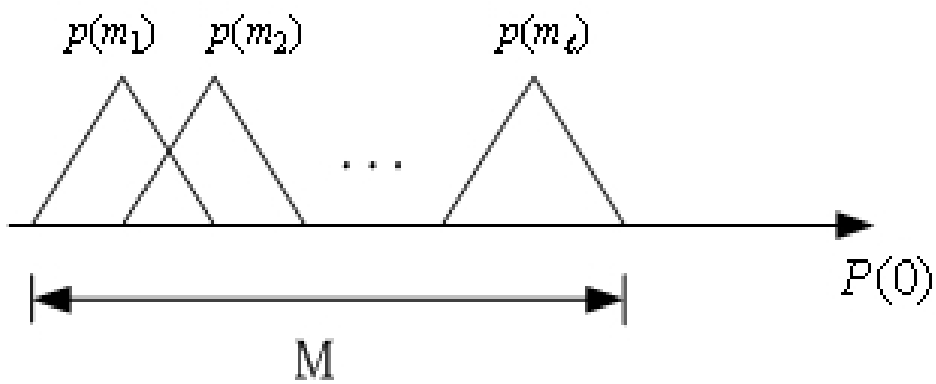
BER Performance vs Interleaver Size (N)



Reduction of Error Coefficients

⋆ Let $P(m_i)$ represent a code path of C_1 encoded from message m_i and let $P\,'(m_i)$ represent the code path of C_2 encoded from message m_i.

⋆ Suppose that $P(m_1), P(m_2), \cdots, P(m_\ell)$ are nearest neighbors of the all zero path $P(0)$.



⋆ Since in RSC a weight 1 message will generate an infinite-weight sequence, hence the weight of m_i is more than 1.

⋆ After the interleaving operation, the more than 1 nonzero bits in m_i will be likely to be separated in a wide range. Hence, it is likely that $P'(m_1), P'(m_2), \cdots, P'(m_\ell)$ will have large weights.

⋆ Let $\frac{M}{N}$ be the error coefficient, where M is the number of nearest neighbors of the all zero codeword of the turbo code.

⋆ Using larger interleaver size can achieve lower coefficient.

⋆ Low error coefficient will result in low error rate.

Recursive Systematic Convolutional Encoders

⋆ We denote the generator matrix for a rate 1/2 RSC code as follows:

$$G(D) = \begin{bmatrix} 1 & \frac{g_2(D)}{g_1(D)} \end{bmatrix} \tag{1}$$

⋆ Observe that, for recursive encoder, the code word will be of finite weight if and only if the input word is divisible by $g_1(D)$.

Corollary 1. A weight-one input word will produce an infinite weight output word.

Corollary 2. For any non-trivial $g_1(D)$, there exists a family of weight-two input words of the form $D^j(1 + D^{q-1})$, $j \geq 0$, which produce finite weight output words. When $g_1(D)$ is a primitive polynomial of degree m, then $q = 2^m$.

Asymptotic Performance for Turbo Codes

⋆ The asymptotic performance of a rate R Turbo code in the additive white Gaussian noise (AWGN) channels with one side power spectrum density N_0 is described as follows.

⋆ Using the standard union bounding technique, the bit error rate (BER) of the Turbo code with maximum-likelihood (ML) decoding can be upper-bounded by

$$P_b \leq \sum_{i=1}^{2^N-1} \frac{w_i}{N} Q(\sqrt{d_i \frac{2RE_b}{N_0}}), \tag{2}$$

where w_i is the weight of the i^{th} message word and d_i is the weight of the i^{th} code word. Please see pp. 532-534 of Lin's book for the derivation of each term in the above equation.

⋆ The above equation can be rewritten as

$$P_b \leq \sum_{w=1}^{N} \sum_{j=1}^{C_w^N} \frac{w}{N} Q(\sqrt{d_{wj} \frac{2RE_b}{N_0}}), \quad (3)$$

where C_w^N is the binomial coefficient and d_{wj} is the weight of the j^{th} code word produced by a weight-w message word.

⋆ Consider the first few terms in the outer summation of equation (3).

w=**1:** ¿From Corollary 1, weight-one message words will produce large weight code words at both constituent encoders. Thus, each d_{1j} is significantly greater than the minimum code words so that the w=1 terms in equation (3) will be negligible.

w=**2:** 1. Of the C_2^N weight-two message words, only a fraction will be divisible by $g_1(D)$ and, of these, only certain one will yield the smallest weight, $d_{2,min}^{CC}$, at a constituent encoder output (here, CC denotes "constituent code").

2. With the interleaver present, if an input $u(D)$ of weight-two

yields a weight-$d_{2,min}^{CC}$ code word at the first encoder's output, it is unlikely that the permuted input, $u'(D)$, seen by the second encoder will also correspond to a weight-$d_{2,min}^{CC}$ code word.

3. However, we can be sure that there will be some minimum-weight turbo code words produced by weight-two message words, and that this minimum weight can be lower-bounded by

$$d_{2,min}^{TC} \geq 2d_{2,min}^{CC} - 2 \equiv d_{f,eff}, \tag{4}$$

where $d_{f,eff}$ is the effective free distance of the Turbo code .

4. The exact value of $d_{2,min}^{TC}$ (here, TC denotes "Turbo code") is interleaver dependent. We may denote the number of weight-two message words which produce weight-$d_{2,min}^{TC}$ turbo code words by n_2 so that, for $w = 2$, the inner sum in

equation (3) can be approximated as

$$\sum_{j=1}^{C_2^N} \frac{2}{N} Q(\sqrt{d_{2j} \frac{2RE_b}{N_0}}) \approx \frac{2n_2}{N} Q(\sqrt{d_{2,min}^{TC} \frac{2RE_b}{N_0}}). \tag{5}$$

w=3: 1. Following an argument similar to the $w = 2$ case, we can approximate the inner sum in equation (3) for $w = 3$ as

$$\sum_{j=1}^{C_3^N} \frac{3}{N} Q(\sqrt{d_{3j} \frac{2RE_b}{N_0}}) \approx \frac{3n_3}{N} Q(\sqrt{d_{3,min}^{TC} \frac{2RE_b}{N_0}}), \tag{6}$$

where n_3 and $d_{3,min}^{TC}$ are obviously defined.

2. While n_3 is clearly dependent on the interleaver, we can make some comments on its size relative to n_2 for a ”randomly generated” interleaver.

   (a) We can expect the number of weight-three terms divisible by $g_1(D)$ to be of the order of the number of weight-two terms divisible by $g_1(D)$. Thus, most of the C_3^N term in equation (3) can be removed from consideration for this reason.

(b) Moreover, given a weight-three encoder input $u(D)$ divisible by $g_1(D)$, it becomes unlikely that the permuted input $u'(D)$ will also be divisible by $g_1(D)$.

(c) For example, suppose $u(D) = g_1(D) = 1 + D + D^2$. Then the interleaver output will be a multiple of $g_1(D)$ if the three input 1's become the j^{th}, $(j+1)^{th}$, and $(j+2)^{th}$ bits out of the interleaver, for some j.

(d) If the interleaver acts in a purely random fashion so that the probability that one of the 1's lands a given position is $1/N$, the interleaver output will be $D^j g_1(D)$ with probability $3!/N^3$. For comparison, for $w = 2$ inputs, a given interleaver output pattern occurs with probability $2!/N^2$. Thus, we can expect the number of weight-three information sequence, n_3, resulting in remergent paths in both encoders to be much less than n_2

$$n_3 \ll n_2, \tag{7}$$

with the result being that the inner sum in equation (3) for $w = 3$ is negligible relative to that for $w = 2$ provided that N is sufficient large.

$w \geq 4$: Using the similar argument, we can show that $n_w \ll n_2$ for $w \geq 4$.

From our discussion above, it is easy to find interleavers such that $w = 2$ term dominates the asymptotic performance of a Turbo code for $N \geq 1000$. Hence, we will use equation (5) to estimate the asymptotic performance of a Turbo code.

Error Performance in the Error-Floor Region

⋆ Let A_d be the number of codewords of weight d and B_d be the total number of nonzero information bits on all weight-d path.

⋆ For a $(k = 1)$ convolutional code,

$$P_b \leq \sum_{d=d_{free}} B_d Q(\sqrt{d\frac{2RE_b}{N_0}}).$$

⋆ Let $\tilde{B}_d = \frac{B_d}{A_d}$. For a turbo code with interleaver size N

$$P_b \leq \sum_{d=d_{free}} \frac{A_d \tilde{B}_d}{N} Q(\sqrt{d\frac{2RE_b}{N_0}}).$$

⋆ There is a rate $\frac{1}{2}$ turbo code with $N = 65536, d_{free} = 6$, $A_6 = 3$, $\tilde{B}_6 = 2$. The free distance asymptote is $P_{d_{free}} = \frac{3\times 2}{65536} Q(\sqrt{6\frac{E_b}{N_0}})$.

⋆ There is a $(2, 1, 14)$ convolutional code with $d_{free} = 18$,$A_{18} = 18$,

$B_{18} = 137$. The free distance asymptote is $P_{free} = 137Q(\sqrt{18\frac{E_b}{N_0}})$

⋆ An "average" turbo code with $N = 65536$ has distance spectrum

d	A_d	B_d
6	4.5	9
8	11	22
10	20.5	41
12	75	150

⋆ The $(2,1,14)$ convolutional code has distance spectrum

d	A_d	B_d
18	33	137
20	136	1034
22	835	7857

Error Performance in the Water-fall Region

⋆ **EXIT** chart is used to explain the dynamics of iterative decoding and to predict the pinch-off signal-to-noise ratios of a turbo code.

⋆ Extrinsic information transfer chart, or **EXIT** chart: Relate a parameter of the input to a constituent decoder to a parameter of the decoder output.

⋆ The parameter can be

1. Input: The signal-to-noise ratio (SNR) of the $a\ priori$ L-value $L_a(u_l)$.
2. Output: The SNR of the $a\ posteriori$ extrinsic L-value $L_e(u_l)$.
3. Input: Mutual information between an information bit u_l and its $a\ priori$ L-value $L_a(u_l)$.
4. Output: Mutual information between an information bit u_l and its $a\ posteriori$ extrinsic L-value $L_e(u_l)$.

EXIT charts based on mutual information

⋆ We model the *a priori* L-value inputs to a constituent decoder as independent Gaussian random variables (r.v.) with variance σ_a^2 and $\mu_a = \pm\sigma_a^2/2$, where the sign of μ_a depends on the transmitted value of u_a based on the following facts.

1. The input channel L-values to a constituent decoder are independent Gaussian r.v. with variance $2L_c$ and mean $\pm L_c$.

2. Extensive simulations of the *a posteriori* extrinsic L-value $L_e(u_l)$ for a constituent decoder with very large block lengths support this assumption.

⋆ The mutual information $I_x[u_l; L_x(u_l)]$ between u_l and $L_x(u_l)$ is

$$\frac{1}{2}\sum_{u_l=-1,+1}\int_{-\infty}^{+\infty} P_{L_x}(\xi|u_l)\log_2\frac{2P_{L_x}(\xi|u_l)}{P_{L_x}(\xi|u_l=-1)+P_{L_x}(\xi|u_l=+1)}d\xi$$

, where x can be either a or e.

Mutual information

1. Mutual information is a measure of the amount of information that one random variable contains about another random variable. It is the reduction in the uncertainty of one random variable due to the knowledge of the other.

2. The mutual information $I(X;Y)$ between two random variables with joint density $f(x,y)$ is defined as

$$
\begin{aligned}
I(X;Y) &= \int\int f(x,y)\log\frac{f(x,y)}{f(x)f(y)}dxdy \\
&= \int\int f(x|y)f(y)\log\frac{f(x|y)}{f(x)}dxdy
\end{aligned}
$$

Mutual information

Let $f(x) = P_{L_x}(\xi)$, $dx = d\xi$,

$$f(y) = P_{u_l}(u_l) = \frac{1}{2}\delta(u_l + 1) + \frac{1}{2}\delta(u_l - 1)$$

$$\Rightarrow I_x[u_l; L_x(u_l)]$$

$$= \int\int P_{L_x}(\xi|u_l)[\frac{1}{2}\delta(u_l + 1) + \frac{1}{2}\delta(u_l - 1)] \log_2 \frac{P_{L_x}(\xi|u_l)}{P_{L_x}(\xi)} d\xi du_l$$

$$= \frac{1}{2}\int P_{L_x}(\xi|u_l = +1) \log_2 \frac{P_{L_x}(\xi|u_l=+1)}{P_{L_x}(\xi)} d\xi +$$

$$\frac{1}{2}\int P_{L_x}(\xi|u_l = -1) \log_2 \frac{P_{L_x}(\xi|u_l=-1)}{P_{L_x}(\xi)} d\xi$$

$$P_{L_x}(\xi) = \sum_{u_l=+1,-1} P_{L_x,u_l}(\xi, u_l)$$

$$= \sum_{u_l=+1,-1} P_{L_x}(\xi|u_l) P_{u_l}(u_l)$$

$$= \frac{1}{2}[P_{L_x}(\xi|u_l = +1) + P_{L_x}(\xi|u_l = -1)]$$

⋆ For a constituent decoder, $P_{L_a}(\xi|u_l)$ is assumed to be Gaussian and $P_{L_e}(\xi|u_l)$ is determined by simulating the BCJR algorithm with independent Gaussian-distributed $a\ priori$ L-value input for a particular constituent code and a large block length.

⋆ Steps for generating decoder input-output transfer curves:

1. Fix a channel signal-to-noise ratio.
2. For a fixed $I_a[u_l; L_a(u_l)]$, run the BCJR algorithm and calculate the associated $I_e[u_l; L_a(u_l)]$.
3. Repeat Step. 2 for different $I_a[u_l; L_a(u_l)]$ and plot the resulting values of $I_e[u_l; L_e(u_l)]$.

⋆ Steps for obtaining **EXIT** chart:

1. Generate and plot decoder input(X-axis)-output(Y-axis) transfer curves for RSC1.
2. Generate and plot decoder input(Y-axis)-output(X-axis) transfer curves for RSC2.

Decoder I/O transfer curves: Various E_b/N_o

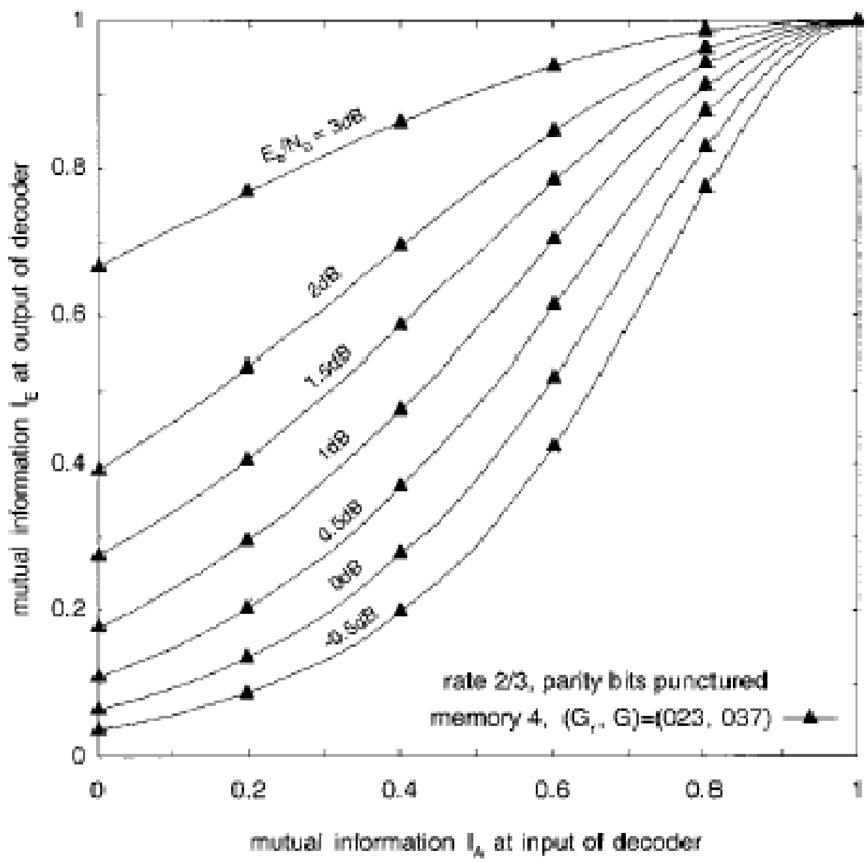


Fig. 2. Extrinsic information transfer characteristics of soft in/soft out decoder for rate 2/3 convolutional code; E_b/N_0 of channel observations serves as parameter to curves.

Decoder I/O transfer curves: Various Component Codes

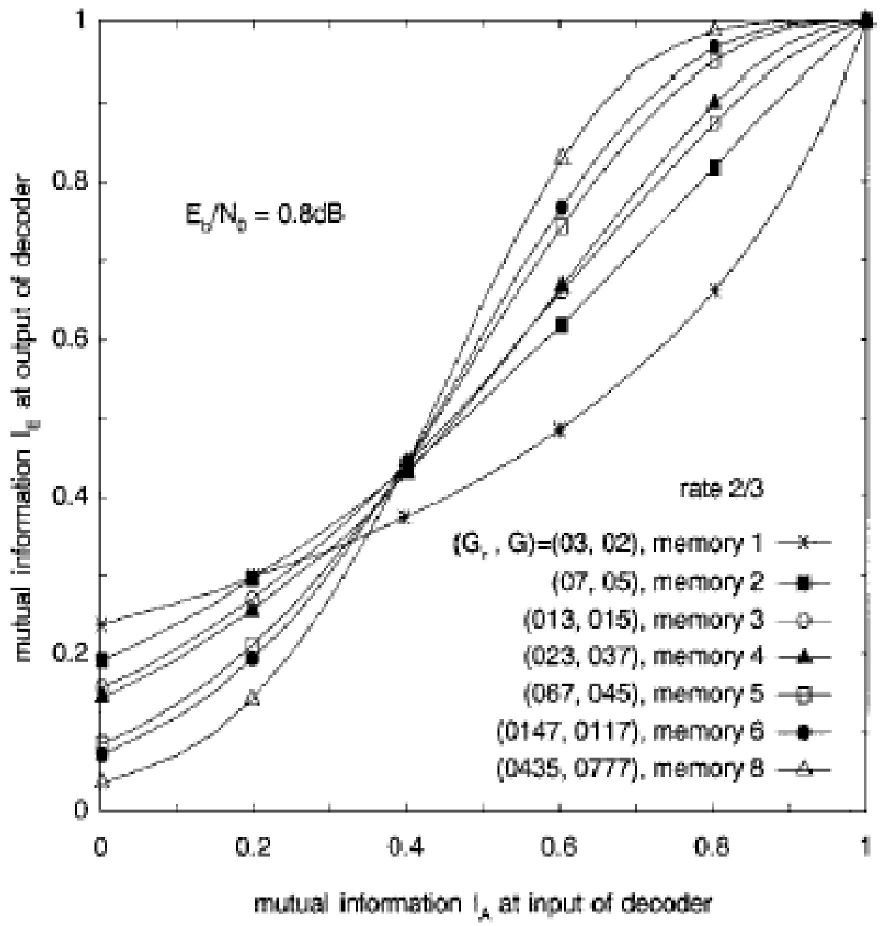


Fig. 3. Extrinsic information transfer characteristics of soft in/soft out decoder for rate 2/3 convolutional code, $E_b/N_0 = 0.8$ dB, different code memory.

EXIT Charts: Pinch-Off SNR limits

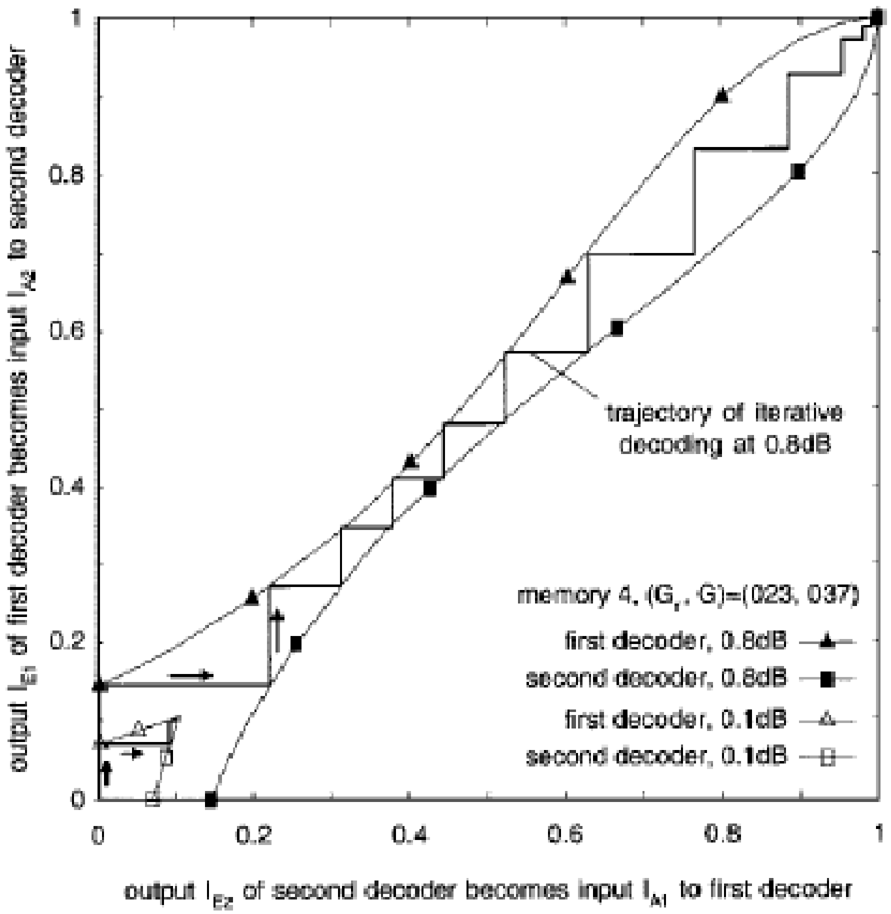
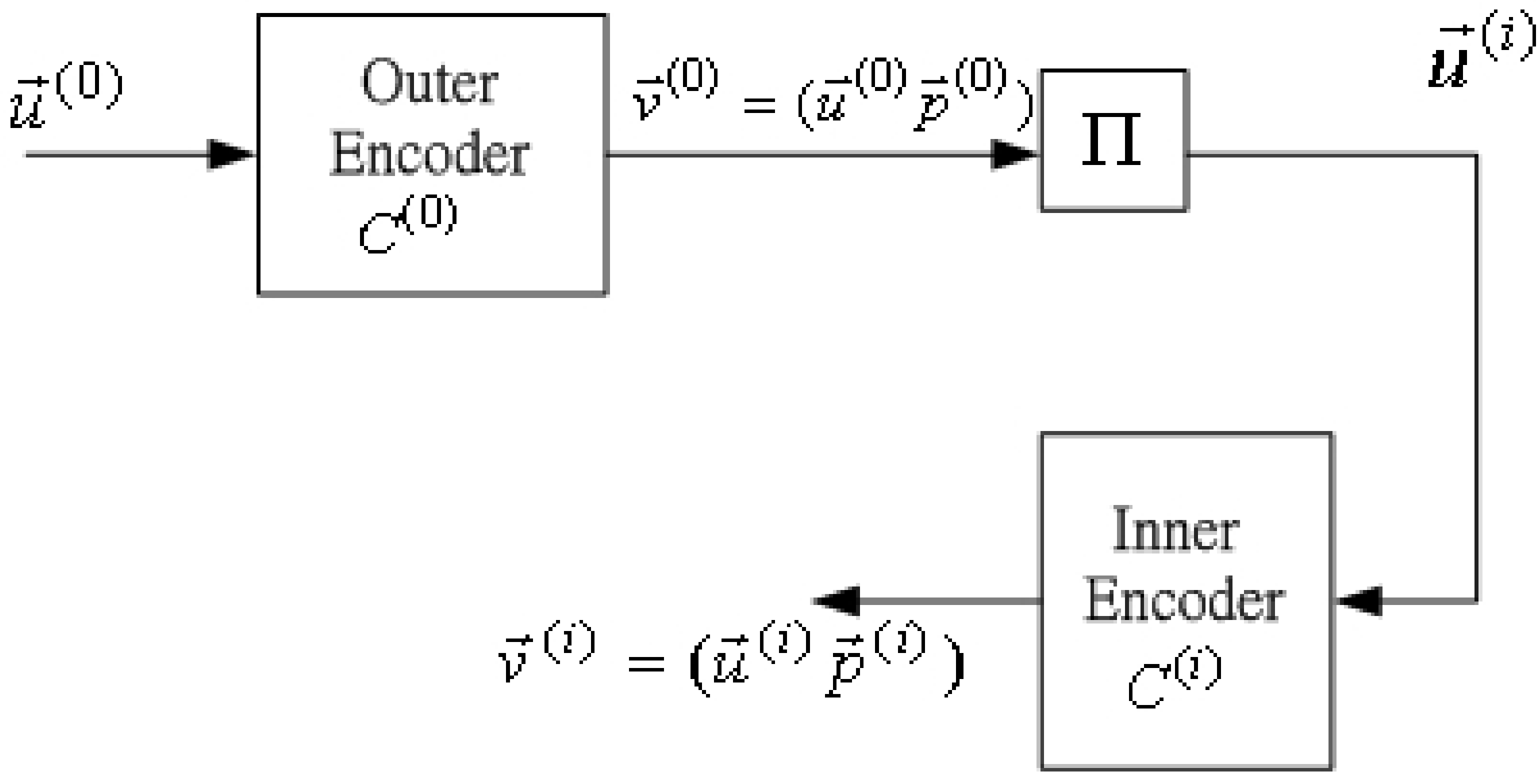


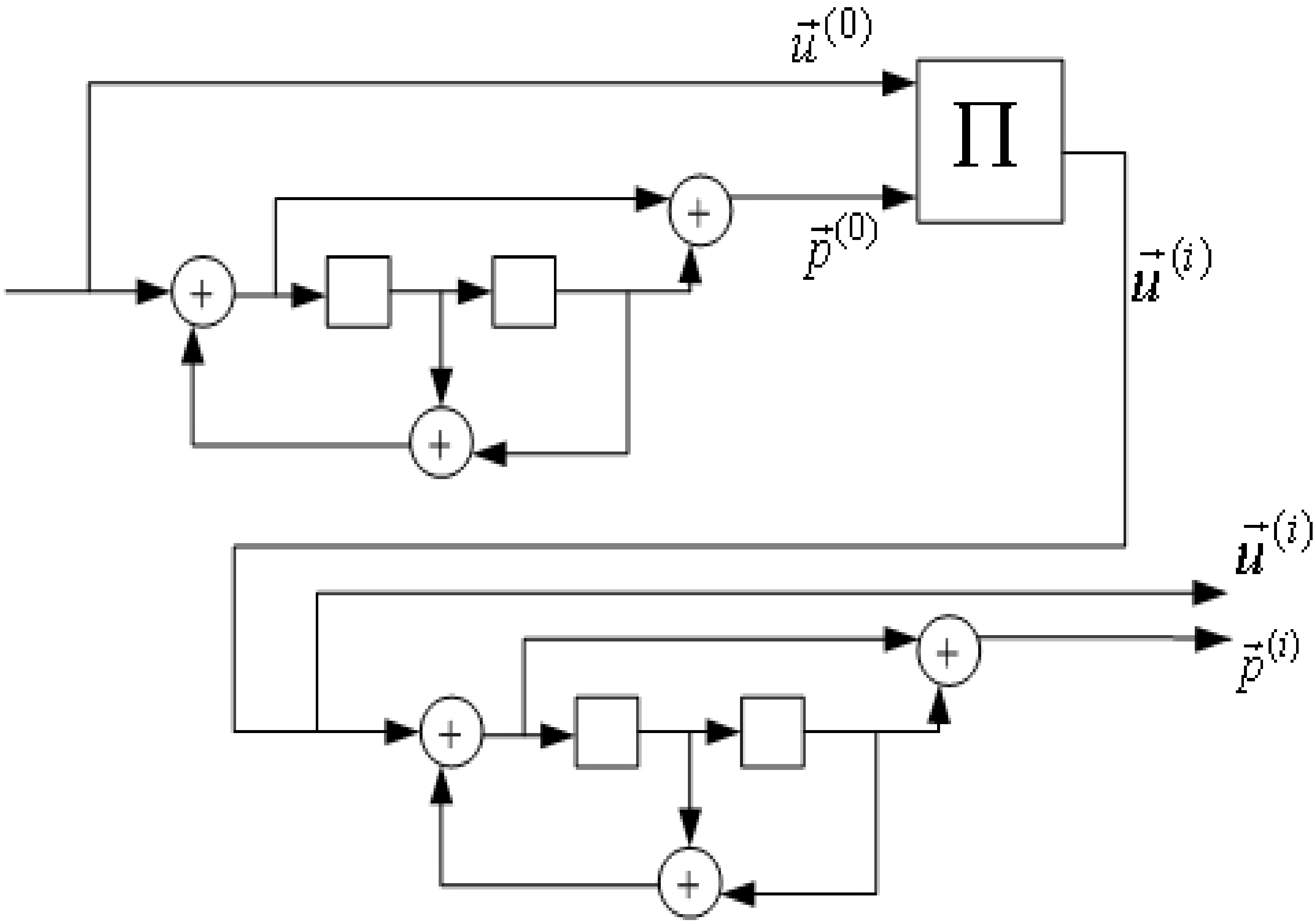
Fig. 5. Simulated trajectories of iterative decoding at $E_b/N_0 = 0.1$ dB and 0.8 dB (symmetric PCC rate 1/2, interleaver size 60 000 systematic bits).

Serial Concatenated Turbo Code

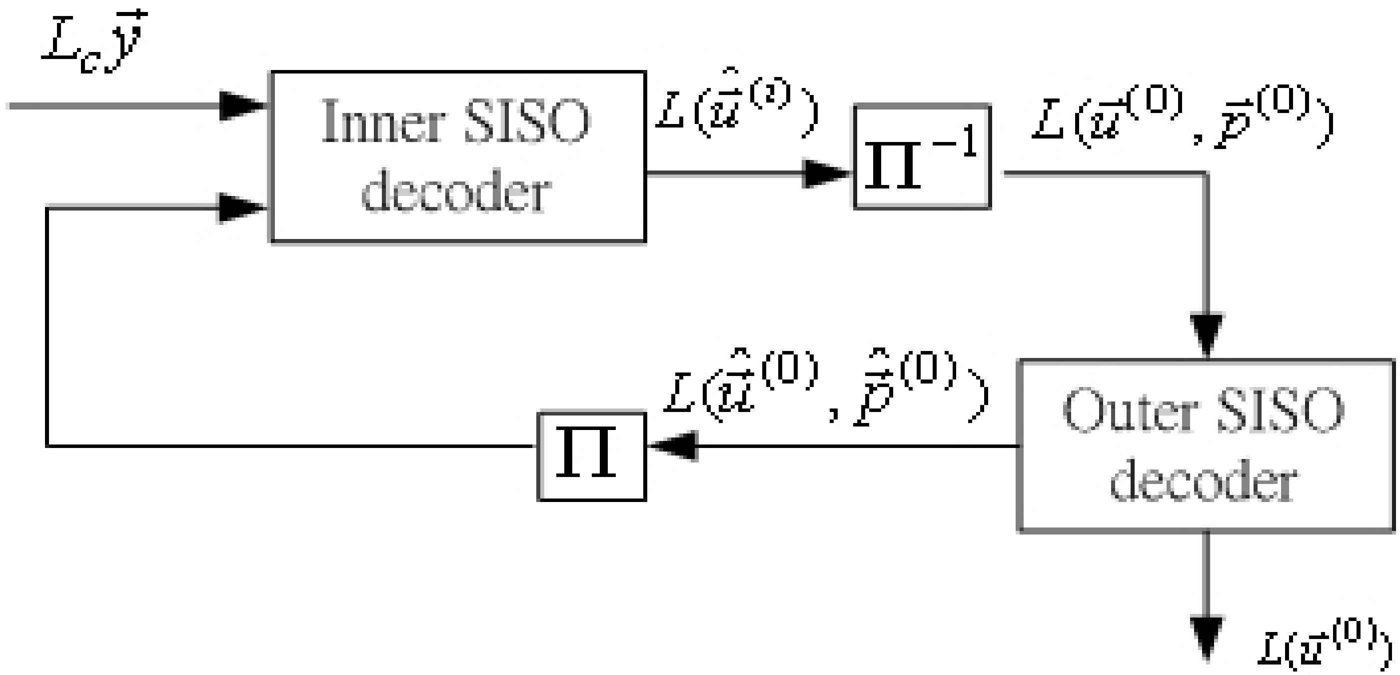
Encoding:



Example 2:



Decoding:



SISO: soft in soft out